EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	12	krzysztof near cwalina.in.	US-PGPUB; USPAT	OR	ON	2007/03/22 15:46
S2	8	bradley near abrams.in.	US-PGPUB; USPAT	OR	ON	2007/03/22 15:48
S3	57	anthony near moore.in.	US-PGPUB; USPAT	OR	ON	2007/03/22 15:49
S4	65	christopher near anderson.in.	US-PGPUB; USPAT	OR	ON	2007/03/22 15:50
S5	19	michael near pizzo.in.	US-PGPUB; USPAT	OR	ON	2007/03/22 15:52
S6	15	robert near3 brigham.in.	US-PGPUB; USPAT	OR	ON	2007/03/22 15:53
S7	15820	microsoft.as.	US-PGPUB; USPAT	OR	ON	2007/03/22 15:53
S8	1	S7 and (((application adj program$5 adj interface) or api) and sample and scenario).clm.	US-PGPUB; USPAT	OR	ON	2007/03/22 15:56
S9	4	S7 and (((application adj program$5 adj interface) or api) and scenario).clm.	US-PGPUB; USPAT	OR	ON	2007/03/22 15:57
S10	800	S7 and ((application adj program$5 adj interface) or api).clm.	US-PGPUB; USPAT	OR	ON	2007/03/22 15:57
S11	113	S7 and (((application adj program$5 adj interface) or api) and language).clm.	US-PGPUB; USPAT	OR	ON	2007/03/22 15:57
S12	32	S7 and (((application adj program$5 adj interface) or api) and language and code).clm.	US-PGPUB; USPAT	OR	ON	2007/03/22 15:58
S13	1500	719/328-329.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:02
S14	4810	717/106-113,124-135.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:04

EAST Search History

S15	2144	(S13 or S14) and ((application adj program$5 adj interface) or api)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:05
S16	1160	S15 and (program$5 near language)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:06
S17	337	S16 and scenario	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:10
S18	128	S17 and sample	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:10
S19	99	S18 and (@pd<"20031023" or @ad<"20031023" or @prad<"20031023" or @rlad<"20031023")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:17
S20	6	("5815703" \| "5966531" \| "6115744" \| "6230117" \| "6256676" \| "6854120").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/03/22 16:20

EAST Search History

S21	89	("20050091660" "6931623" "20010025372" "20050187930" "20050193291" "20060248206" "20070011274" "20070011275" "6401081" "20060230348" "6041365" "20050144593" "6292830" "5732270" "6349343" "20050160414" "6125391" "6226675" "6226675" "6301557" "6928648" "7069260" "7089560" "7096474" "20020156937" "20020156938" "20030217128" "20050203771" "6356957" "20010016881" "5956506" "6185590" "6425017" "6434447" "6775834" "20020124119" "20030163479" "20030236859" "20040148612" "20050193266" "20060104306" "20060168122" "20050160432" "5838972" "20060150200" "5710908" "5899990" "6199195" "6330659" "7152228").pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:38
S22	12	((application adj program$5 adj interface) or api) and (code adj scenario)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:44
S23	10258	((application adj program$5 adj interface) or api) near3 (design$4 or creat$4 or implement$5 or generat$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:45
S24	1782	S23 and ("717".clas. or "719".clas.)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:45
S25	1015	S24 and (program$4 adj language)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:45

EAST Search History

S26	40	S25 and (code adj sample)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:46
S27	12	("5036484" \| "5604896" \| "5619685" \| "5623657" \| "5627979" \| "5634053" \| "5812768" \| "5845589" \| "6263497" \| "6330711" \| "6382846" \| "6438744").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/03/22 16:49
S28	2	("5987247").PN.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:51
S29	2	("20020083212").PN.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/22 16:51
S31	1	wo-200188702-$.did.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 08:30
S32	2	"6842892".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 08:31
S34	2	"20010052111".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 08:35
S35	17	("5546584" \| "5675805" \| "5748974" \| "5761502" \| "5875332" \| "5907847" \| "5920718" \| "5931919" \| "6076092" \| "6134545" \| "6173290" \| "6182277" \| "6230157" \| "6285976" \| "6321374" \| "6341371" \| "6557100").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/03/26 08:42

EAST Search History

S36	12	("5857100"\|"5887172"\|"6167565"\|"6282579"\|"6484309"\|"6698014"). PN.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 08:54
S38	2	"5732270".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:05
S39	2	"5097533".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:05
S40	63	("3665421" \| "4493027" \| "4667290" \| "4787035" \| "4905138" \| "4961133").PN. OR ("5097533"). URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/03/26 09:06
S41	771	design$4 near (api or (application adj progam$4 adj interface))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:16
S42	563	S41 and language	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:16
S43	157	S42 and (refin$4 or revis$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:17
S44	134	S43 and (@pd<"20031023" or @ad<"20031023" or @prad<"20031023" or @rlad<"20031023")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:09

EAST Search History

S45	267	S41 and (program$4 adj language)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:34
S46	3	S45 and ((refin$4 or revis$4) with api)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:36
S47	95	S45 and ("717".clas. or "719".clas.)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 09:36
S48	89	(refin$4 or revis$4 or correct$4) near api	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 10:37
S49	71	S48 and (@pd<"20031023" or @ad<"20031023" or @prad<"20031023" or @rlad<"20031023")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 10:32
S50	40	(refin$4 or revis$4) near api	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 10:37
S51	40	(refin$4 or revis$4) near (api or (application adj progam$4 adj interface))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 10:38
S52	166	(refin$4 or revis$4) near3 (api or (application adj progam$4 adj interface))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 10:38

EAST Search History

S53	4	S52 and ((api or (application adj progam$4 adj interface)) with (complex or complicat$4))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 10:39
S54	23	("5247681" \| "5375241" \| "5379432" \| "5404528" \| "5410698" \| "5432925" \| "5437036").PN. OR ("6006279").URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/03/26 10:41
S55	2344	test$4 near3 (api or (application adj progam$4 adj interface))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 10:53
S56	2349	(test$4 or benchmark$4) near3 (api or (application adj progam$4 adj interface))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:09
S57	123	S56 and ("717".clas. or "719".clas.)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:09
S58	101	S57 and (@pd<"20031023" or @ad<"20031023" or @prad<"20031023" or @rlad<"20031023")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:17
S59	441	usability adj (test$4 or stud$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:16
S60	143	S59 and (api or (application adj progam$4 adj interface))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:16

EAST Search History

S61	118	S60 and (programmer or developer)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:16
S62	115	S61 and (@pd<"20031023" or @ad<"20031023" or @prad<"20031023" or @rlad<"20031023")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:17
S63	128	(performance or usability or benchmark$4) near (api or (application adj program$4 adj interface))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:32
S64	44	S63 and (programmer or developer)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:22
S65	6	(performance or usability or benchmark$4) near (api or (application adj program$4 adj interface)) with (programmer or developer)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:33
S66	374	(programmer or developer or user) with (test$4 or benchmark$4) with (api or (application adj program$4 adj interface))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:36
S67	51	S66 and "717".clas.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/26 11:36

PORTAL USPTO
Subscribe (Full Service) Register (Limited Service, Free) Login
Search: The ACM Digital Library The Guide
"application programming interface" api design SEARCH
THE ACM DIGITAL LIBRARY
Feedback Report a problem Satisfaction survey

Terms used **application programming interface api design** Found **16,185** of **198,991**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 A java API for historical ciphers: an object-oriented design project
Ralph Morelli, Ralph Walde, Gregg Marcuccio
February 2001 **ACM SIGCSE Bulletin , Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education SIGCSE '01**, Volume 33 Issue 1
Publisher: ACM Press
Full text available: pdf(185.90 KB) Additional Information: full citation, abstract, references, index terms

This paper describes a project suitable for a software engineering or object-oriented design course. The project consists of asking students to design an *application programming interface (API)* for a particular range of applications. An API-design project has several features not always found in application-design projects: It forces students to focus carefully on the distinction between the programming and the user interfaces; it provides a good justification for studying existing APIs a ...

2 Knowledge sharing in software engineering: Sometimes you need to see through walls: a field study of application programming interfaces
Cleidson R. B. de Souza, David Redmiles, Li-Te Cheng, David Millen, John Patterson
November 2004 **Proceedings of the 2004 ACM conference on Computer supported cooperative work CSCW '04**
Publisher: ACM Press
Full text available: pdf(209.03 KB) Additional Information: full citation, abstract, references, citings, index terms

Information hiding is one of the most important and influential principles in software engineering. It prescribes that software modules hide implementation details from other modules in order to decrease the dependency between them. This separation also decreases the dependency among software developers implementing modules, thus simplifying some aspects of collaboration. A common instantiation of this principle is in the form of application programming interfaces (APIs). We performed a field ...

Keywords: application programming interfaces, collaborative software development, interfaces, qualitative studies

3 Mathematics of computing: A Java API for polynomial arithmetic
Claire Whelan, Adam Duffy, Andrew Burnett, Tom Dowling
June 2003 **Proceedings of the 2nd international conference on Principles and practice of programming in Java PPPJ '03**
Publisher: Computer Science Press, Inc.
Full text available: pdf(145.78 KB) Additional Information: full citation, abstract, references, index terms, review

This paper looks at the development of a Java application programming interface (API) for performing unbounded Polynomial Arithmetic. It shows how Java can be used to perform large integer mathematical operations by using the BigInteger class. By demonstrating how the API might be used in a real application the paper shows how easy the API is to use with very little knowledge of polynomial arithmetic.

Keywords: API, Java, arithmetic, polynomial

4 Design of a high performance volume visualization system

Barthold Lichtenbelt
August 1997 **Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware HWWS '97**
Publisher: ACM Press
Full text available: pdf(1.11 MB) Additional Information: full citation, references, citings, index terms

Keywords: OpenGL, system design, visualization, volume accelerator, volume rendering

5 Software evolution: CatchUp!: capturing and replaying refactorings to support API evolution

Johannes Henkel, Amer Diwan
May 2005 **Proceedings of the 27th international conference on Software engineering ICSE '05 , Proceedings of the 27th international conference on Software engineering ICSE '05**
Publisher: ACM Press, IEEE Computer Society
Full text available: pdf(298.59 KB) Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

Library developers who have to evolve a library to accommodate changing requirements often face a dilemma: Either they implement a clean, efficient solution but risk breaking client code, or they maintain compatibility with client code, but pay with increased design complexity and thus higher maintenance costs over time.We address this dilemma by presenting a lightweight approach for evolving application programming interfaces (APIs), which does not depend on version control or configuration man ...

Keywords: application programming interfaces, refactoring, software evolution

6 Session 16: special session: MPSOC design tools: Programming models and HW-SW interfaces abstraction for multi-processor SoC

Ahmed A. Jerraya, Aimen Bouchhima, Frédéric Pétrot
July 2006 **Proceedings of the 43rd annual conference on Design automation DAC '06**
Publisher: ACM Press
Full text available: pdf(671.83 KB) Additional Information: full citation, abstract, references, index terms

For the design of classic computers the Parallel programming concept is used to abstract HW/SW interfaces during high level specification of application software. The software is then adapted to an existing multiprocessor platforms using a low level software layers that implement the programming model. Unlike classic computers, the design of heterogeneous MPSoC includes also building the processors and other kind of hardware components required to execute the software. In this case, the programm ...

Keywords: HW/SW interfaces, heterogeneous MPSoC, programming models

7 Architecting in the Face of Uncertainty: An Experience Report

Ian Gorton, Jereme Haack

May 2004 **Proceedings of the 26th International Conference on Software Engineering ICSE '04**
Publisher: IEEE Computer Society
Full text available: pdf(218.04 KB) Additional Information: full citation, abstract, citings, index terms

Understanding an applicationýs functional and non-functionalrequirements is normally seen as essential fordeveloping a robust product suited to client needs. Thispaper describes our experiences in a project that, bynecessity, commenced well before concrete clientrequirements could be known. After a first version of theapplication was successfully released, emergingrequirements forced an evolution of the applicationarchitecture. The key reasons for this are explained,along with the architectural ...

8 Compiling SpecC for simulation
Jianwen Zhu, Daniel D. Gajski
January 2001 **Proceedings of the 2001 conference on Asia South Pacific design automation ASP-DAC '01**
Publisher: ACM Press
Full text available: pdf(59.21 KB) Additional Information: full citation, abstract, references, index terms

Systems-on-chip (SOC) design calls for the use of executable system design language (SLDL). SpecC is a C-based SLDL designed to embrace the IP-centric design methodology. In this paper, we present a SpecC-language-based approach for system level simulation of SOC. Pros and cons of this approach is compared against the existing library-based approaches. Furthermore, we discuss in detail the various design considerations for the SpecC simulation API as well as our reference implementatio ...

9 Automatically connecting documentation to code with rose
Robert Pierce, Scott Tilley
October 2002 **Proceedings of the 20th annual international conference on Computer documentation SIGDOC '02**
Publisher: ACM Press
Full text available: pdf(256.32 KB) Additional Information: full citation, abstract, references, citings, index terms

One of the most common problems with program documentation is keeping it synchronized with the source code it purports to explain. One solution to this problem is to automate the documentation process using reverse engineering technology. Reverse engineering is an emerging branch of software engineering that focuses on recreating high-level information (such as program documentation) from low-level artifacts (such as source code). This paper describes an automated approach to maintaining the con ...

Keywords: application programming interface (API), automation, documentation, single sourcing, software engineering

10 Experiences in Design and Implementation of a High Performance Transport Protocol
Yunhong Gu, Xinwei Hong, Robert L. Grossman
November 2004 **Proceedings of the 2004 ACM/IEEE conference on Supercomputing SC '04**
Publisher: IEEE Computer Society
Full text available: pdf(193.37 KB) Additional Information: full citation, abstract, citings

This paper describes our experiences in the development of the UDP-based Data Transport (UDT) protocol, an application level transport protocol used in distributed data intensive applications. The new protocol is motivated by the emergence of wide area high-speed optical networks, in which TCP is often found to fail to utilize the abundant bandwidth. UDT demonstrates good efficiency and fairness (including RTT fairness and TCP friendliness) characteristics in high performance computing applicati ...

Keywords: Performance, Design, Experimentation, UDT, transport protocol, data intensive application, design, implementation

11 Chip hierarchical design system (CHDS): a foundation for timing-driven physical design into the 21st century

R. G. Bushroe, S. DasGupta, A. Dengi, P. Fisher, S. Grout, G. Ledenbach, N. S. Nagaraj, R. Steele

April 1997 **Proceedings of the 1997 international symposium on Physical design ISPD '97**

Publisher: ACM Press

Full text available: pdf(860.02 KB) Additional Information: full citation, references, citings, index terms

12 IO-Lite: a unified I/O buffering and caching system

Vivek S. Pai, Peter Druschel, Willy Zwaenepoel

February 2000 **ACM Transactions on Computer Systems (TOCS)**, Volume 18 Issue 1

Publisher: ACM Press

Full text available: pdf(196.15 KB) Additional Information: full citation, abstract, references, citings, index terms

This article presents the design, implementation, and evaluation of IO -Lite, a unified I/O buffering and caching system for general-purpose operating systems. IO-Lite unifies all buffering and caching in the system, to the extent permitted by the hardware. In particular, it allows applications, the interprocess communication system, the file system, the file cache, and the network subsystem to safely and concurrently share a single physical copy of the data. Protection and ...

Keywords: I/O buffering, caching, networking, zero-copy

13 Client-server computing

Alok Sinha

July 1992 **Communications of the ACM**, Volume 35 Issue 7

Publisher: ACM Press

Full text available: pdf(7.53 MB) Additional Information: full citation, references, citings, index terms, review

Keywords: client-server computing

14 Middleware: a model for distributed system services

Philip A. Bernstein

February 1996 **Communications of the ACM**, Volume 39 Issue 2

Publisher: ACM Press

Full text available: pdf(238.25 KB) Additional Information: full citation, references, citings, index terms

15 Empirical methods and program understanding: Work experience versus refactoring to design patterns: a controlled experiment

T. H. Ng, S. C. Cheung, W. K. Chan, Y. T. Yu

November 2006 **Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering SIGSOFT '06/FSE-14**

Publisher: ACM Press

Full text available: pdf(396.39 KB) Additional Information: full citation, abstract, references, index terms

Program refactoring using design patterns is an attractive approach for facilitating anticipated changes. Its benefit depends on at least two factors, namely the effort involved in the refactoring and how effective it is. For example, the benefit would be small

if too much effort is required to translate a program correctly into a refactorized form, and whether such a form could effectively guide maintainers to complete anticipated changes is unknown. A metric of effectiveness is the maintainers ...

Keywords: controlled experiment, design patterns, refactoring

16 Embedded software: Fast and accurate timed execution of high level embedded software using HW/SW interface simulation model

Aimen Bouchhima, Sungjoo Yoo, Ahmed Jeraya

January 2004 **Proceedings of the 2004 conference on Asia South Pacific design automation: electronic design and solution fair ASP-DAC '04 , Proceedings of the 2004 conference on Asia South Pacific design automation: electronic design and solution fair ASP-DAC '04**

Publisher: IEEE Press

Full text available: pdf(180.07 KB) Publisher Site Additional Information: full citation, abstract, references, citings

In this paper, we propose a methodology to perform early design stage validation of hardware/software (HW/SW) systems using a HW/SW interface simulation model. Given a SW application described at the OS abstraction level and a HW Platform described at an arbitrary abstraction level, we aim at providing the adaptation layer, i.e. simulation model of the HW/SW interface, which will enable the timed HW/SW cosimulation of the entire system at an early design stage before the system design is complet ...

17 Technical papers: Grid resource management---Designing a runtime system for volunteer computing

David P. Anderson, Carl Christensen, Bruce Allen

November 2006 **Proceedings of the 2006 ACM/IEEE conference on Supercomputing SC '06**

Publisher: ACM Press

Full text available: pdf(168.15 KB) html(2.13 KB) Additional Information: full citation, abstract, references

Volunteer computing is a form of distributed computing in which the general public volunteers processing and storage to scientific research projects. BOINC, a middleware system for volunteer computing, is currently used by about 20 projects, to which 300,000 volunteers and 450,000 computers supply 350 TeraFLOPS of processing power. A BOINC client program runs on the volunteered hosts and manages the execution of applications. Together with a library linked to applications, it implements a ...

18 Triangles: design of a physical/digital construction kit

Matthew G. Gorbet, Maggie Orth

August 1997 **Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques DIS '97**

Publisher: ACM Press

Full text available: pdf(545.55 KB) Additional Information: full citation, references, citings, index terms

Keywords: collaboration, connections, digital connector, interface design, physical interface

19 Patterns: MAPO: mining API usages from open source repositories

Tao Xie, Jian Pei

May 2006 **Proceedings of the 2006 international workshop on Mining software repositories MSR '06**

Publisher: ACM Press

Full text available: pdf(79.25 KB) Additional Information: full citation, abstract, references, index terms

To improve software productivity, when constructing new software systems, developers often reuse existing class libraries or frameworks by invoking their APIs. Those APIs, however, are often complex and not well documented, posing barriers for developers to use them in new client code. To get familiar with how those APIs are used, developers may search the Web using a general search engine to find relevant documents or code examples. Developers can also use a source code search engine to search ...

Keywords: application programming interfaces, mining software repositories, program comprehension

20 Embedded systems: MDA-based approach for embedded software generation from a UML/MOF repository

Francisco A. M. do Nascimento, Marcio F. da S. Oliveira, Marco A. Wehrmeister, Carlos E. Pereira, Flávio R. Wagner

August 2006 **Proceedings of the 19th annual symposium on Integrated circuits and systems design SBCCI '06**

Publisher: ACM Press

Full text available: pdf(341.15 KB) Additional Information: full citation, abstract, references, index terms

This work presents a UML/MOF meta-modeling infrastructure, following the MDA approach, which is oriented to the design space exploration of embedded systems, based on the independent specification of application and platform. A mapping from the application into the platform is implemented by model transformations. A meta-data repository implementation is also introduced, automatically providing an Application Programming Interface (API) that is suitable to the manipulation of the UML meta-models ...

Keywords: MDA, UML, design space exploration, embedded systems design

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

Sign in

Google

Web Images Video News Maps more »

api design Search Advanced Search Preferences

Web

Results 1 - 10 of about 125,000,000 for api design. (0.30 seconds)

openide: How To Design a (module) API
Hints for best practices when designing module APIs to make them powerful, readable, and maintainable with compatibility.
openide.netbeans.org/tutorial/api-design.html - 88k - Cached - Similar pages

Java API Design Guidelines
There are tons of books and articles about how to design and write good Java code, but surprisingly little about the specific topic of API design. ...
www.artima.com/weblogs/viewpost.jsp?thread=142428 - 34k - Cached - Similar pages

Interface Design: Best Practices in Object-Oriented API Design in Java
A work in progress book about object-oriented Java design, written by Bill Venners.
www.artima.com/interfacedesign/ - 18k - Cached - Similar pages
[More results from www.artima.com]

[PDF] How to Design a Good API and Why it Matters
File Format: PDF/Adobe Acrobat - View as HTML
Why is API Design Important? •. APIs can be among a company's greatest assets ...
Effects of API Design Decisions on. Performance are Real and Permanent ...
lcsd05.cs.tamu.edu/slides/keynote.pdf - Similar pages

How to design Good APIs and Why they Matter
Plenary session on 3/31 - how to design a good API - and why it matters. ... Methodology for API design - there are a few things that tend to lead to good ...
www.cincomsmalltalk.com/blog/blogView?showComments=true&entry=3258158706 - 48k - Cached - Similar pages

InfoQ: How to Design a Good API & Why it Matters
A well-written API can be a great asset to the organization that wrote it and to all that use it. Given the importance of good API design, ...
www.infoq.com/presentations/effective-api-design - 49k - Mar 25, 2007 - Cached - Similar pages

Java Collections API Design FAQ
This is the most controversial design decision in the whole API. Clearly, static (compile time) type checking is highly desirable, and is the norm in Java. ...
java.sun.com/j2se/1.5.0/docs/guide/collections/designfaq.html - 19k - Cached - Similar pages

The Old Joel on Software Forum - API Design
I'm heading up the API design for my company's next gen software package and was wondering if anyone knows of some good resources that deal with designing ...
discuss.fogcreek.com/joelonsoftware/default.asp?cmd=show&ixPost=15878 - 10k - Cached - Similar pages

Joshua Bloch: A conversation about design - Java World
Bill Venners: In the preface of your fine book, Effective Java Language Programming Guide, you write that you tend to think in terms of API design. ...
www.javaworld.com/javaworld/jw-01-2002/jw-0104-bloch_p.html - 41k - Cached - Similar pages

Amazon.com: Inside Java 2 Platform Security: Architecture, API ...
Amazon.com: Inside Java 2 Platform Security: Architecture, API Design, and Implementation (2nd Edition): Books: Li Gong,Gary Ellison,Mary Dageforde by Li ...

www.amazon.com/Inside-Java-Platform-Security-Implementation/dp/0201787911 - 136k - Mar 25, 2007 - Cached - Similar pages

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

Download Google Pack: free essential software for your PC

api design Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

©2007 Google